● Group 39

April 28, 2024

CS335 Project

Python 3.8 Compiler

Acknowledgement

We would like to thank our instructor Mr. Swarnendu Biswas for teaching us the course theory which was directly applicable in this project.

We would like to thank our TA Ms. Lavanya for giving feedback through the milestones.

Milestone 1

- **Implementation Tools**
  - Lexer - flex
  - Parser - bison
  - Implementation Language - C++

Milestone 1

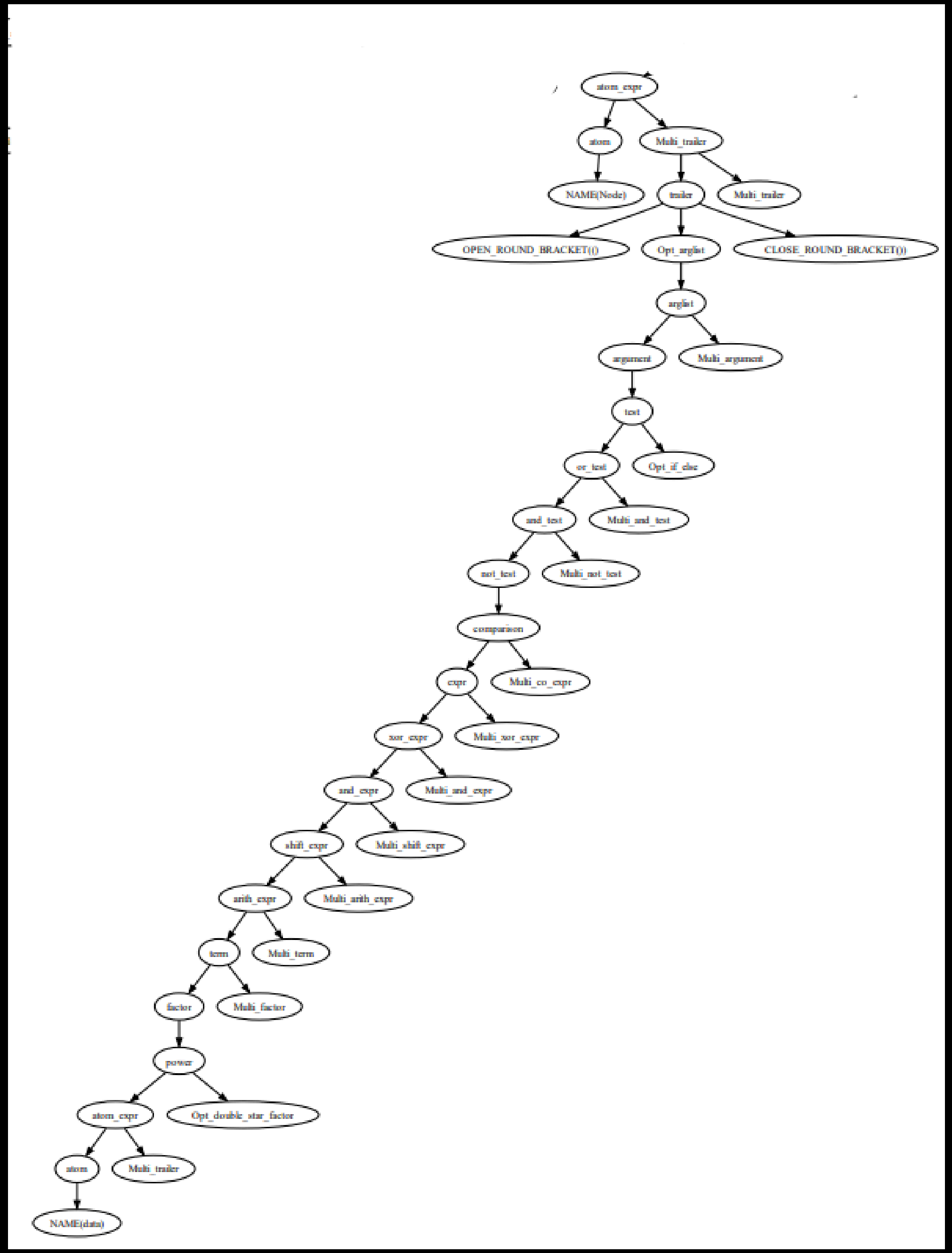
- **Lexer**
  - Identified necessary tokens
  - Wrote regular expressions for all tokens
    Relatively hard ones: Long String, Short String, INDENT, DEDENT, NEWLINE
  - Used a stack and exclusive state (%x) for INDENT and DEDENT tokens
  - Used a stack and inclusive state (%s) to support implicit line joining

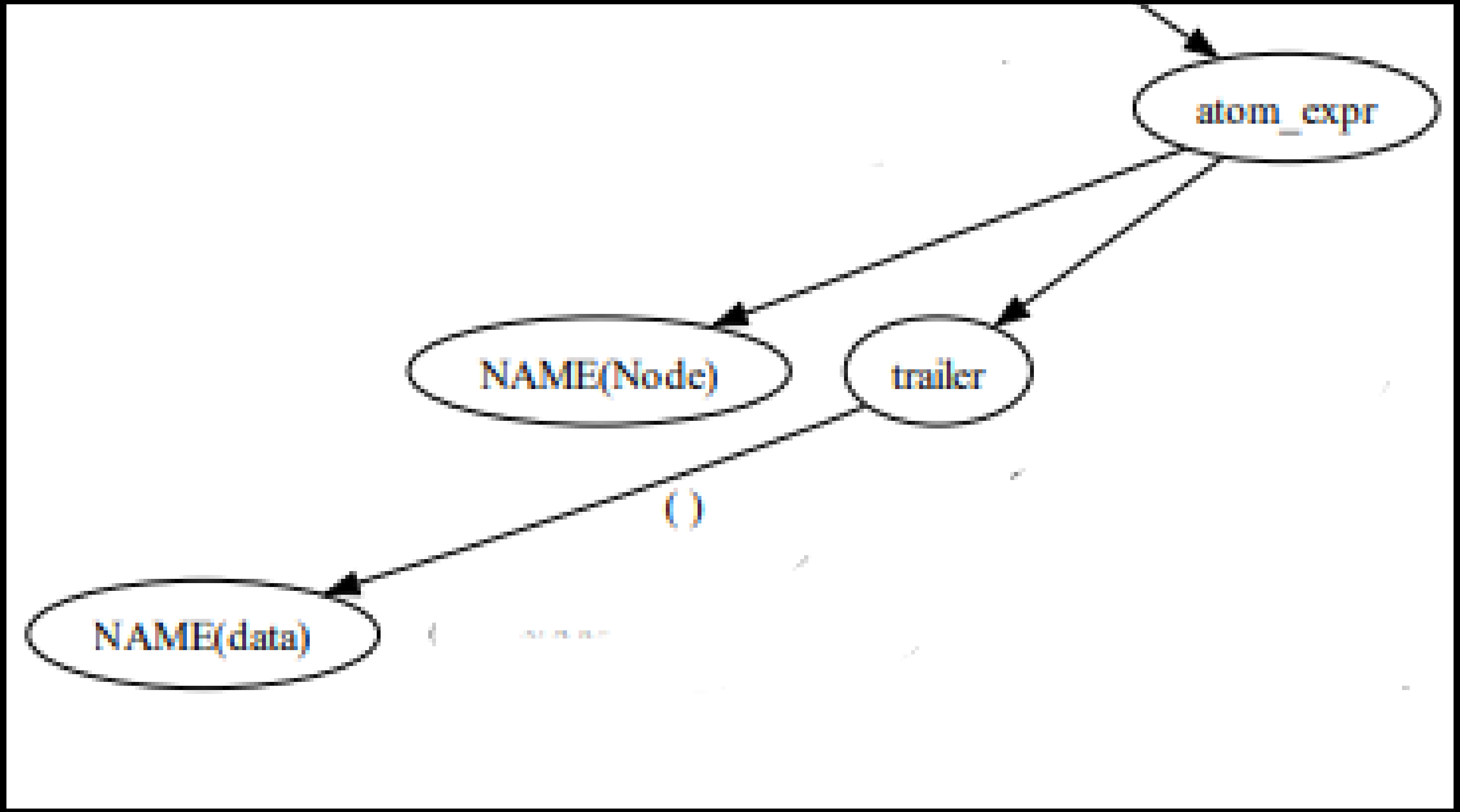
Milestone 1

- **Changes to Grammar (Python 3.8)**
  - Removed all non-required tokens and non-terminals from grammar
  - Changed productions with operators from right recursive to left recursive
  - Removed epsilon productions causing shift-reduce conflicts
  - Introduced extra non-terminals to incorporate the regular expressions given in standard grammar to Bison

Milestone 1

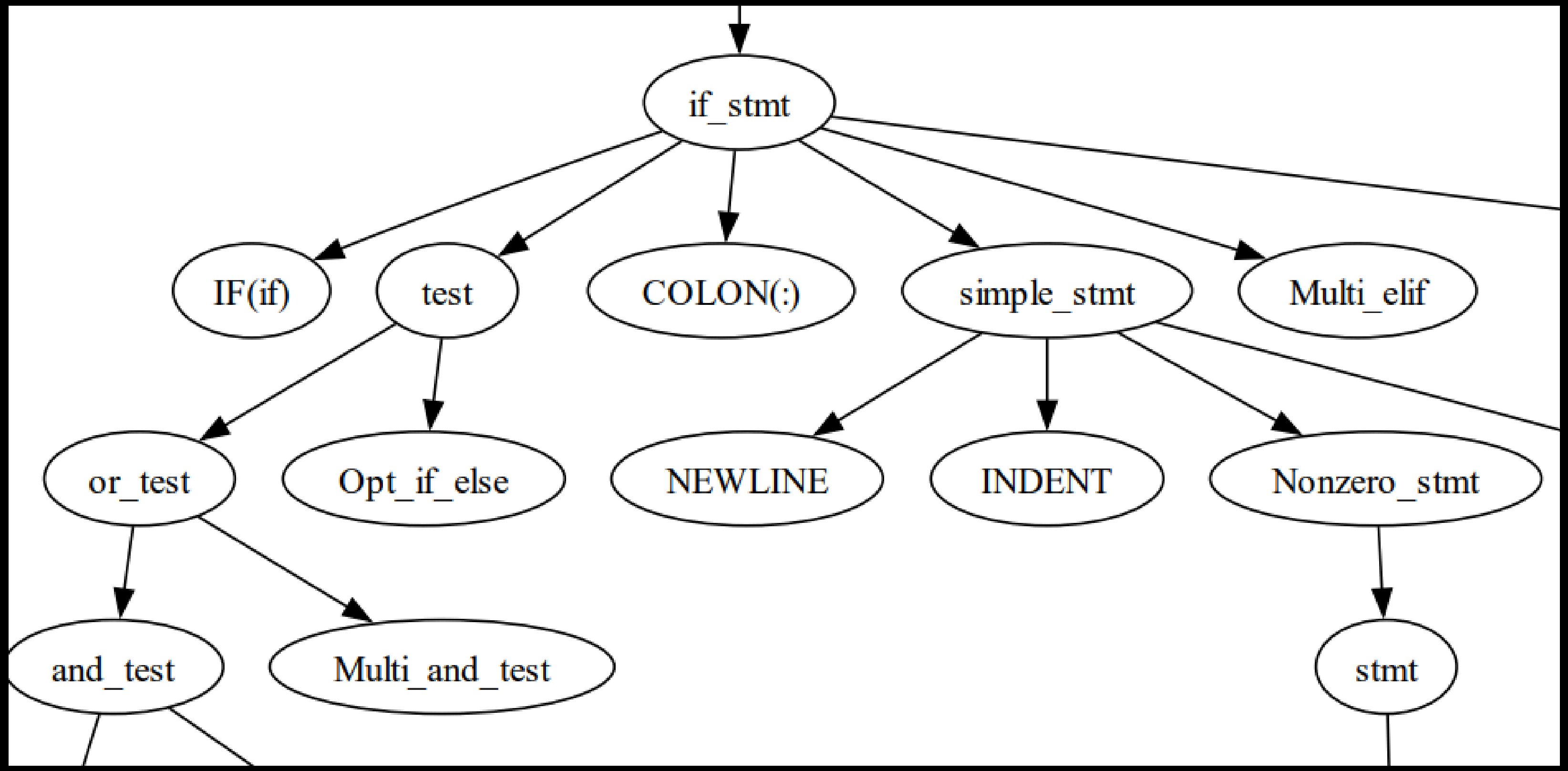
- **Construction of AST**
  - Removed unnecessary intermediate non-terminals

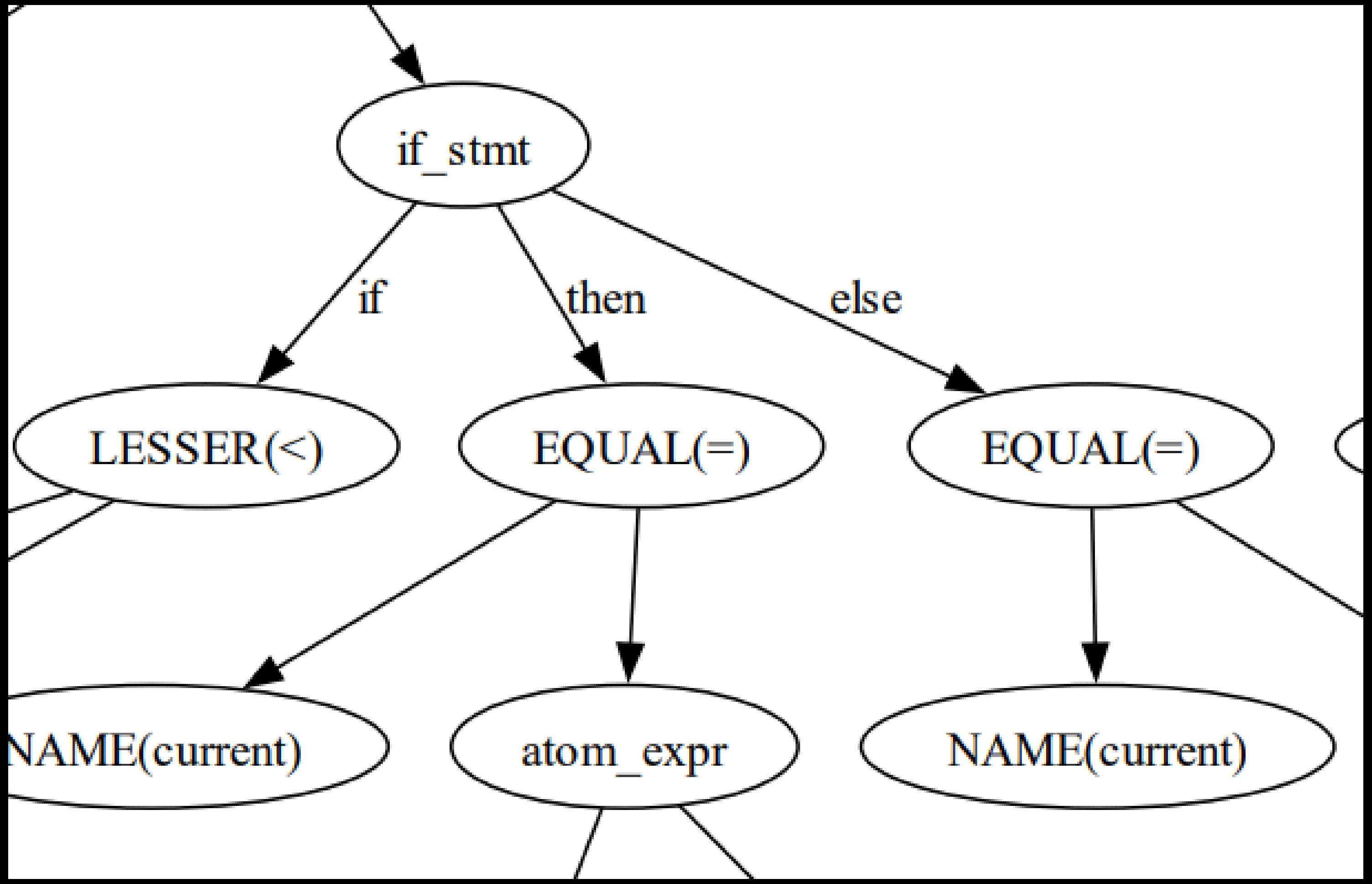




Milestone 1

- **Construction of AST**
  - Labelled the edges and removed non-semantic nodes for better readability





Milestone 1

- **Construction of AST**
    - Made operator as parent node and the operands as its children node
    - Removed unnecessary tokens from AST like NEWLINE, INDENT, DEDENT, COMMA, SEMICOLON as they are implicitly implied

Milestone 2

- **Separating traversal path in AST of l-values and r-values in expression statements by modifying grammar**
  - **Why is this separation needed?**
  - Every string of terminals derivable from non-terminal "test" cannot be l-value. (e.g.:   2*b = 4;   3 = y)
  - We need to detect uninitialized variable error only in r-values of expression statement
  - In l-value, we further divide into l-value-decl and l-value-nondecl. **decl** means that it can be declared. (e.g.:   a : int, self.a : bool). **nondecl** means that it cannot be declared (e.g.:   obj.attr : int,   list[idx] : float).
  - Separation helps in imposing different constraints for l-values and r-values

Milestone 2

- **Populating Symbol Table**
    - Token, type/returntype, offset, scope, initialization status, number of parameters, parameter number
    - For class and function symbol table entries, we store the pointer to symbol table of that class and function in the entry
    - All symbol tables contain pointers to parent symbol table
    - Class Inheritance: We copy parent class‘ symbol table entries to child class‘ symbol table before proceeding to populate it with further child class entries

Milestone 2

- **Type Checking and Typecasting**
    - All assignment statements
    - All operation statements
    - Function parameters and arguments (including “self” parameter)
    - Function return values

Milestone 2

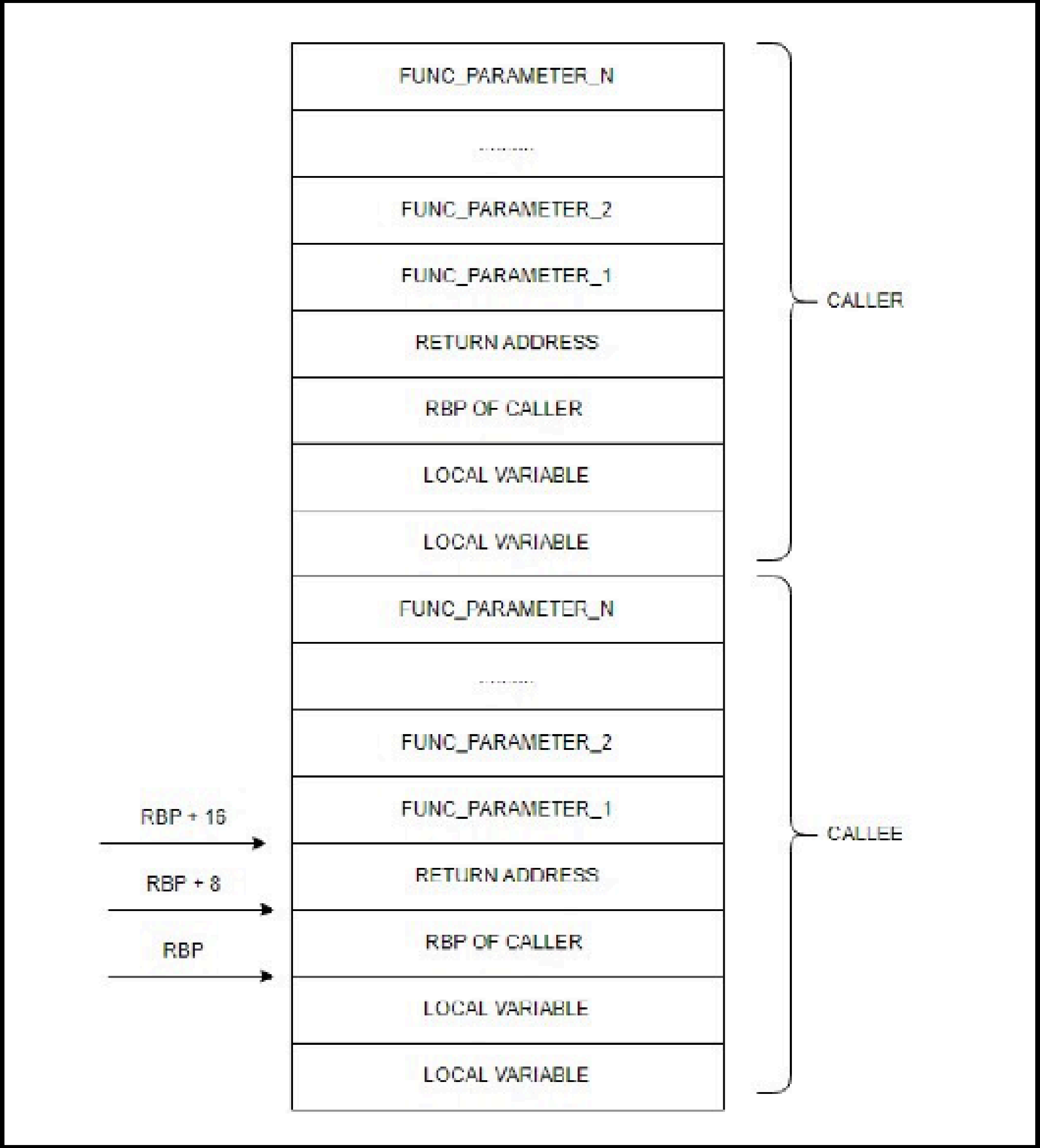
- **Error Handling** (reported at **160+** places in the source code)
  - Redclaration in same scope
  - Redeclaration of global variable in any scope
  - Type errors
  - Invalid assignments
  - Wrong number of arguments passed
  - Wrong type of arguments passed
  - Undeclared function/method/variable/attribute/class/object
  - **Unitialized variable/attribute**
  - Constructor call
  - Invalid operator for types of operands
  - List index type must be “int“
  - Index error, for indexing non-list types or for indexing lists with higher dimensions

Milestone 2

- **3AC Generation**
  - Does not bother with error handling. **gen()** statements are written after error checking is done.
  - New temporaries for every new node in AST for operation productions
  - “for”, “if”, “while”, “break”, “continue” are separately dealt with to generate appropriate labels and goto instructions
  - Lists and objects are stored as pointers pointing to heap memory. Their elements are stored in this memory and referenced through the pointer

Milestone 2

- **Runtime Support**
  - Manipulation of %rsp and %rbp is shown in 3AC
  - Stack architecture diagram



Milestone 3

- **Function Definition (Callee Activation Record)**
  - Push old %rbp
  - Push callee saved registers
  - Decrement %rsp depending on the number of variables and temporaries used in the function
  - Parse all the statements in function body suite. Retrieve parameters of the function by positive offsets to %rbp
  - Store the return value (if it exists) in %rax
  - Increment %rsp pointer
  - Pop callee saved registers
  - Return to caller

Milestone 3

- **Function Call (Caller Activation Record)**
  - Align the stack so that %rsp is at multiple of 16 bytes. This depends upon if #arguments of function being called is odd or even
  - Push caller saved registers
  - Push the arguments
  - Call the function (Pushes the return address into the stack)
  - Retrieve the return value (if it exists) from %rax
  - Increment %rsp pointer depending on #arguments
  - Pop caller saved registers
  - Increment %rsp by 8 if the stack was aligned before function call

Milestone 3

- **PLT Functions**
    - Align the stack before calling PLT functions
    - strcmp@plt, malloc@plt, printf@plt
- **Strings**
    - Stored in section RODATA
- **Typecasting**
    - Internally implemented bool <--> int typecasting in asm
- **len()** and **range()**
    - Internally implemented range() function in for statements
    - Stored length of the list as the first element in the heap memory allocated to the list

Milestone 3

- **Heap Memory Allocation**
  - Lists - Allocated (length of the list + 1)*8 bytes in heap memory using malloc@plt for lists. First element is length of the list
  - Objects - Allocated size of the object in heap memory using malloc@plt

Hard Features Supported

- **Method calling by both class and object**
  - obj.method(arg1, arg2)
  - class.method(obj, arg1, arg2)

- **obj.attr[idx]** fully supported in both r-values and l-values

- **len()** function internally implemented and can be called even if the exact list itself is not known (in a function whose parameter is list)

*No unsupported features

Team Members

- Sawan H N - 210952 - 33%
- Shrilakshmi S K - 211012 - 33%
- Yashas D - 211199 - 33%

Thank You